AD-A214 882

[redacted]

Axiomatix

SIMULATION AND EMULATION OF DYNAMIC JAMMING OF THE MOBILE SUBSCRIBER EQUIPMENT (MSE)

Interim Technical Report

Unjeng Cheng
Gaylord K. Huth

November 6, 1989

U. S. Army Research Office
P. O. Box 12211
Research Triangle Park, NC 27709-2211

Contract No. DAAL 03-87-C-0007

Axiomatix
9841 Airport Boulevard
Suite 1130
Los Angeles, CA 90045

APPROVED FOR PUBLIC RELEASE
DISTRIBUTION LIMITED

Axiomatix Report No. R8911-1

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ARO 24649.7-EL-S	2. GOVT ACCESSION NO. NA	3. RECIPIENT'S CATALOG NUMBER NA
4. TITLE (and Subtitle) SIMULATION AND EMULATION OF DYNAMIC JAMMING OF THE DYNAMIC JAMMING OF THE MOBILE SUBSCRIBER EQUIPMENT (MSE)		5. TYPE OF REPORT & PERIOD COVERED Interim 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Unjeng Cheng and Gaylord K. Huth		8. CONTRACT OR GRANT NUMBER(s) DAAL03-87-C-0007
9. PERFORMING ORGANIZATION NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NA
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Research Office P. O. Box 12211 Research Triangle Park, NC 27709-2211		12. REPORT DATE November 6, 1989 13. NUMBER OF PAGES 19
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

NA

18. SUPPLEMENTARY NOTES

The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Adaptive Networks, Code-Division-Multiple-Access (CDMA), Communications, Networks, Network Simulation, Network Emulation, Dynamic Jamming, Packet Radio, Spread Spectrum, Mobile Subscriber Equipment (MSE).

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A Mobile Subscriber Equipment (MSE) test bed would include a simulator and an emulator. The simulator is based on models closely resembling the operation of the network and is designed for statistical analysis and for understanding the general behavior of the MSE network in the jamming environments. The emulator is designed for network operational tests and is used to emulate the real field test of the MSE network under various jamming attacks. The simulator is based on the appropriate statistics associated with the network, but the emulator is capable of processing the real data.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE 10

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

1. INTRODUCTION

The survivability of a military communication network under jamming attack is an important issue. The jamming attack can be static or dynamic. Static jammers stay on for a long time whereas, dynamic jammers can change between the on and off states frequently. It is easier to detect, locate, and destroy the static jammers than dynamic jammers. Moreover, static jammers do not impose a serious threat since adaptive routing algorithms can always circumvent the jammed area; whereas, the dynamic jammers can harm the adaptive routing algorithms more severely. In order to understand and assess the impact of the dynamic jamming threat, we need an appropriate network simulator and emulator with the dynamic jamming simulation and emulation capabilities, respectively. The vulnerable points of a communication network depends on its architecture. For instance, a Packet Radio Network (PRnet) can be attacked on its network, link, and physical layers. A Mobile Subscriber Equipment (MSE) network can be attacked on its flood search algorithm, affiliation process, and radio protocol. The behavior of PRnets under the attack of an intelligent jammer were investigated before [1]. A PRnet simulator with the dynamic jamming simulation capability was developed [1]. It is possible to extend our existing PRnet simulation capability to study the MSE network under the intelligent jamming attack.

A MSE test bed would include a simulator and an emulator. The simulator is based on models closely resembling the operation of the network; whereas, the emulator resembles the network itself. The simulator is designed for statistical analysis and for understanding the general behavior of the MSE network in the jamming environments. The emulator is designed for network operational tests and is used to emulate the real field test of the MSE network under various jamming attacks. The simulator is based on the appropriate statistics associated with the network, but the emulator is capable of processing the real data. The emulator is desirable for verification of the network functions and the simulator is needed for performance evaluation of the network. We must emphasize that there is no clear boundary between simulation and emulation. We can have a system which

performs simulation of certain network operations and also performs emulation of the other network operations. Fortunately, the development of the simulator and emulator are not separate tasks. By planning carefully, their development would be incremental efforts. The software architecture itself is based on the incremental development concept. The MSE simulator would be developed on top of our existing PRnet simulator such that the effort and cost are minimized. Furthermore, the MSE emulator would be developed on top of the MSE simulator.

The software is designed to run on the IBM PC, PC/XT, PC/AT, or compatible computers with an appropriate graphic display. It will have an extensive context-sensitive help system as well as comprehensive documentation. The software will be easy to use and have a friendly user interface. The C programming language would be used for the main body of the simulator and emulator.

The object-oriented programming methodology would be used for the simulator and emulator. This approach can make the software resemble the real system very closely. The hardware components are treated as objects in the simulator. The objects can be divided into many classes. The objects in the same class share the same set of service routines (a layer of the simulator or emulator). An object can contain other objects. This concept resembles the real system where a complex component comprises many simpler components. In this manner, an object can contain as much information as its real hardware counterpart contains. In addition, each object also contains the necessary information for simulator control. When performing simulation, the objects for the low level hardware are described by their mathematical models; whereas, when performing emulation, those objects are described in terms of their corresponding hardware implementations. The operations of the higher level objects, which contains the lower level objects, are independent of how the lower level objects are implemented. This concept allows the incremental development of the proposed MSE simulator and emulator. Therefore, we can start with a simulator which emulates most of the high level protocols

such as the flood search algorithm and the affiliation process, but only simulates the low level functions such as the channel characteristics. We can later extend the simulator to the emulator by emulating the low level functions in detail.

The software architecture and data structure allows us to use the incremental development methodology, namely, the simulator and emulator can be developed in several phases. The products would have appropriate version numbers. The newer version products are the extension of the older version ones.

Our existing PRnet simulator is described briefly in Section 2. A MSE network simulator is described in Section 3.1 and a MSE emulator is described in Section 3.2. The summary is given in Section 4.

2. A PACKET RADIO NETWORK SIMULATOR WITH DYNAMIC JAMMING SIMULATION CAPABILITY

The Packet Radio Network (PRnet) simulator has a layered architecture. Its functions are divided into four major layers: (1) user-interface facilities, (2) simulation software, (3) file server, and (4) supporting library. The user interface design of the software is aimed at wide circulation of the software. It has many user-friendly features. An extensive help system is provided so that the novice users can use the package without any difficulties.

The user-interface layer provides the essential functions for simulator control. It can be further divided into three sublayers: (1) network editor, (2) parameter entry facilities, and (3) result presentation facilities. The common functions needed in the network simulation tasks are identified and are developed as a ready-to-use library for the future use. For instance, a network editor is essential to all kinds of network simulation tasks. The editor displays the network in color graphics and lets the operators add nodes, delete nodes, move nodes, add links, and delete links. We packaged the editor as a ready-to-use routine. The library will have extensive documentation. The user-interface facilities are developed with the off-shelf development tools; thus, the efforts are minimized.

The simulation layer can be further divided into four sublayers (1) jamming simulator, (2) node operations and routing simulation, (3) channel simulator, and (4) input traffic simulation. The jamming simulator generates the jamming-signal-to-noise ratio at each node in each slot according to the prescribed jamming strategy. The jamming strategy can be created and saved in a file, and it can be loaded into memory during simulation. The node operations include (1) collecting the external and transient packets, (2) transmitting the packets in the queue using the prevailing routing table, and (3) generating the "I-AM-HERE" packets if needed. The routing simulation includes (1) transmitting information exchange packets (2) channel quality monitoring, and (3) updating the routing table. The channel simulator determines whether a transmitted packet is received correctly based on a

mathematical model taking into account both multiple access and direct-sequence spread spectrum. The input traffic simulator generates new external arrival packets. For high arrival rates, the uniform random number generator is used in simulation; whereas, for low arrival rates, the geometric random number generator is used.

Using the simulator, we were able to show that the worst case dynamic jamming strategy depends on the network topology, traffic distribution, network information exchange interval, and channel monitoring scheme. We demonstrated in emulation the process that after the PRnet routing algorithm circumvents the jammed nodes, the worst case dynamic jammer changes the jamming pattern accordingly to block the new routes.

3. A MSE NETWORK SIMULATOR AND EMULATOR WITH DYNAMIC JAMMING SIMULATION CAPABILITY

MSE is not jam-resistant. Therefore, it is essential to understand the network behavior under a dynamic jamming attack. The dynamic jamming threat is to destroy the network routing functions. MSE uses the flood algorithm to find the circuit from source to destination. To set up a circuit, the source node center (NC) or large extension node (LEN) switch first broadcasts a search message, which is further forwarded by every switch receiving it which is not the destination. It takes finite amount of time for the search message to reach the destination switch. After receiving the search message, the destination switch responds with a connection message. It also takes finite amount of time for the connection message to return to the source switch. In dynamic jamming environments, the jamming pattern during the route search phase and that during the route setup phase can be different. This creates a serious problem because the route constructed in the route search phase may not be valid in the route setup phase.

Dynamic jamming can also attack the radio protocol. MSE uses the per-call frequency selection process to reduce vulnerability to jamming and interference because blocked or jammed channels are treated as if being used and are ignored in the call-initiation process. Each radio continuously monitoring the quality of the received signal. When the signal drops below threshold for a sufficiently long duration, the channel is marked as jammed. In dynamic jamming environments, a channel can be good in one time interval and be bad in another time interval. Thus, a channel can be selected by the frequency selection process, but it turns bad thereafter. On the other hand, a channel can be precluded by the frequency selection process, but it becomes good later.

The above observations illustrate the importance of understanding the dynamic jamming threat on the MSE network. A MSE simulator would be the extension of our existing PRnet simulator. The required incremental development to upgrade our existing PRnet simulator for MSE application is investigated in Section 3.1. The MSE emulator

would be the extension of the MSE simulator. The required incremental development to implement the emulator will be investigated in Section 3.2.

3.1 A MSE Simulator with Dynamic Jamming Simulation Capability

The MSE simulator simulates the critical functions of MSE. It has two modes: emulation mode and statistical analysis mode. In the emulation mode, it can show various network operations in progress. The speed of emulation is definable by the operators. In the statistical analysis mode, operators can collect various performance statistics. The results can be displayed as the simulation is in progress or they can be accessed by interrupting the simulation temporarily. The architecture, functions, and implementation of the simulator are described in the following:

Software Architecture:

The software has a layered architecture and is highly modularized. The simulator would comprise six layers:

(1) simulator control,
(2) subscriber services and traffic generation,
(3) NC/LEN flood search algorithm and affiliation process,
(4) radio protocol,
(5) dynamic jamming, and
(6) channel simulation.

The architecture is illustrated in Figure 1. The highest layer is the simulator control layer which includes the components such as the MSE network editor, the file server, and the data entry facilities. The simulator control layer influences the operations of all the lower layers via the appropriate control parameters. The second layer contains two components, i.e., subscriber services and traffic generation. The traffic generation simulates the input traffic to the network and the subscriber service layer simulates the services provided by MSE. The third layer simulates the flood search algorithm and affiliation process. The fourth layer simulates the radio protocol. The lowest layers are the dynamic jamming simulation and channel simulation layers.

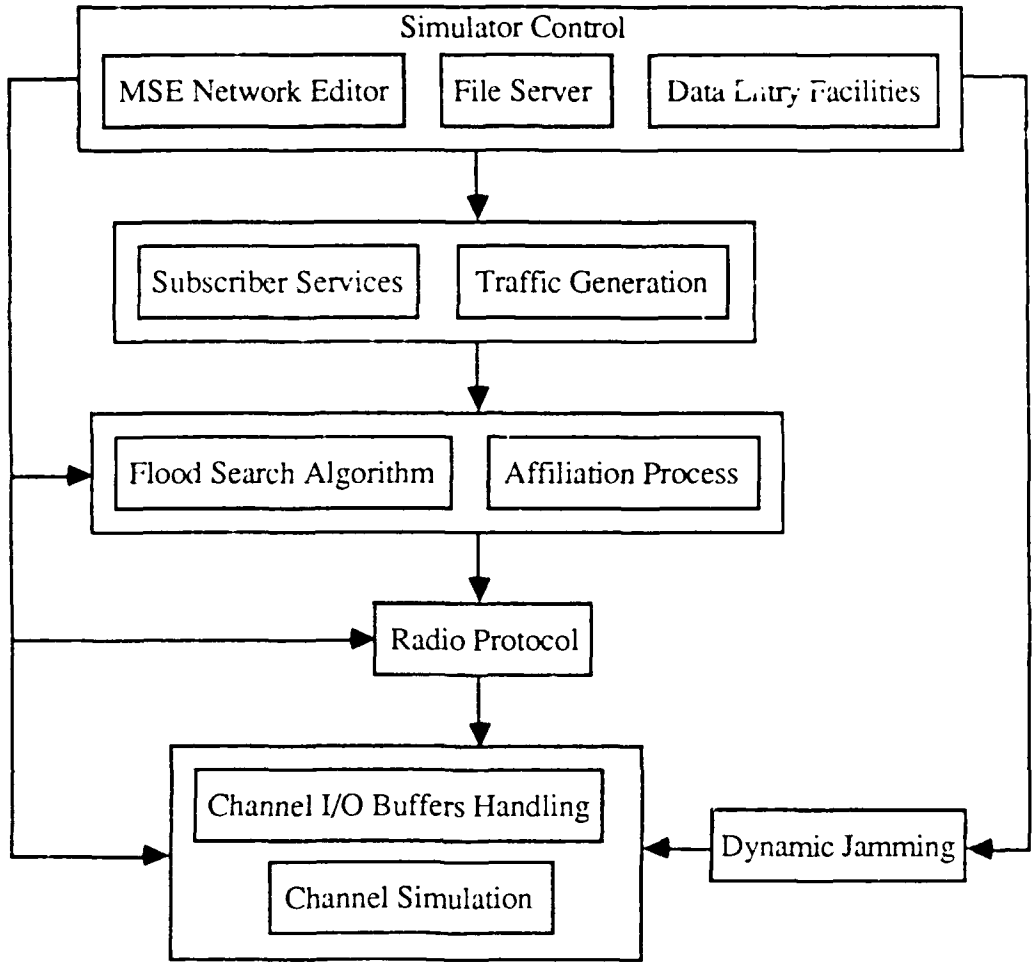

Figure 1. The Architecture of the MSE Simulator.

In the simulator, a node control block is allocated to each node. The control block contains the following information:

- Node identification number: An identification number is assigned uniquely to every node in MSE network. When editing the network, the number is assigned as a node is added and is removed as a node is deleted.

- Component Type: The component can be a subscriber terminal, a mobile subscriber radio-telephone terminal (MSRT), a large extension node (LEN) switch, an small extension node (SEN) switch, a node center (NC) switch, a radio access unit (RAU), or a system control center (SCC). There are four types of subscriber terminals, i.e., digital subscriber voice terminal (DSVT), digital nonsecure voice telephone (DNVT), digital facsimile terminal (FAX), and signal subscriber terminal (SST). The component type determines the type of information to be stored in the node control block, the way to use the

information for MSE simulation, and the operations to be performed by the lower layer software.

- Affiliation Table: For the NC/LEN switches, this field is used to store the affiliation information such as the directory of the affiliated units. For a subscriber terminal, this field is used to store its own dialing numbers
- Neighbor Table: This field is used by the NC/LEN switch to store the information about its neighbor NC/LEN switches.
- Blacklist Table: This field is used by the NC/LEN switches to store the blacklist of the rejected subscriber terminals.
- Channel Usage Table: This filed is used by NC/LEN/SEN/RAU to store the channel allocation information.
- Traffic Load Control: This field is used by the NC/LEN switches for the precedence control.
- Handling Type: This field is used to determine the type of operations to be performed by the simulator or emulator. For instance, you can specify whether the channels should be simulated or be emulated, i.e., the message tokens or the real data should be processed.
- Search Message Buffer: This buffer is used by NC/LEN switches to store the received search messages before their respective circuit setup requests are completed.
- Channel Input/Output Buffers: The I/O buffers are the interface between the higher layer network functions and the channel simulation or emulation layer. The types of data queued in the I/O buffers depend on whether the channel is simulated or emulated.
- Graphical Data: The data stored in this field are for displaying the node in color graphics. The information such as the position, color, and shape to be used for the node are specified here

Function Descriptions:

The simulator control layer includes the components such as the network editor, the file server, and the data entry facilities. The network editor lets operators draw the MSE network. Each type of nodes is represented by its respective type of icons. Each type of links is represented by its respective line types. The colors of displaying objects are used to enhance the user interface. For instance, different colors can be used for various precedence control levels. The data entry facilities let operators create and maintain (1) simulation configuration, (2) traffic generation descriptor, (3) affiliation simulation

descriptor, (4) dynamic jamming descriptor, and (5) channel simulation control parameters. The file server lets operators save and retrieve the simulator control data described above as well as the simulation output.

The subscriber service emulation lets operators choose the desired subscriber terminal, affiliate it with network if necessary, and place the phone call. No voice data will flow through the emulation network. However, operators can specify how long the call will take. The circuit for the call will be held for the requested amount of time. The service request and control messages are represented by their respective tokens containing only the necessary information for appropriate network actions. The usage of message tokens saves the computation time and makes the statistical analysis feasible.

Automatic traffic generation is used for simulation study. The traffic input to the network is simulated according to the prescribed traffic distribution statistics, which can be obtained from previous experience and projected traffic data. The traffic simulation is controlled by the traffic generation descriptor, which can be static or dynamic. For the static case, the traffic statistics do not vary with time; whereas, for the dynamic case, the traffic statistics can vary with time. The dynamic features allow simulation of the scenario from non-combat situations to combat situations and vice versa. The traffic generation descriptor can be saved in a file, and it can be loaded into the memory before simulation. An editor is provided for creating and maintaining the descriptor data.

The status of the flood search algorithm and that of the affiliation process are completely described in its node control block. The flood search algorithm is implemented as a routine used for all NC/LEN switches. To perform the flood search functions at each NC/LEN switch, the simulator simply calls the flood search routine with the node control block of the desired switch being the input. Basically, the flood search routine performs the following functions at each switch:

- If the called party is affiliated at this switch, it issues a connection message back toward the originating switch over the marked routing path; otherwise, it

forwards the received search message to all neighbor NC/LEN switches and the path is marked.

- If the calling party is affiliated at this switch, it issues an end-of-routing messages to all neighbor NC/LEN switches; otherwise, it forwards the received connection message back toward the originating switch over the marked routing path.

The affiliation process contains two types of operations. The first type of operations affiliates MSRTs with RAUs and lets RAUs and SENs forward their affiliation information to the appropriate NC/LEN switches. This type of operations also takes care of the detail affiliation procedure as any subscriber terminal generates the affiliation request. The operations run automatically. The second type of operations lets the subscriber terminals issue their affiliation requests. The operations can be manual or automatic. For the manual application, operators can choose their desired subscriber terminal and issue the affiliation request for that terminal. In addition to the manual scenario, there is also an automatic affiliation request generator which generates the requests according to the affiliation generation descriptor. Combining the manual and automatic operations, operators are able to test each individual affiliation request while MSE is driven by the automatic affiliation generation. The affiliation descriptor data can be saved in a file and can be loaded completely into memory before the simulation if there is enough memory, or it can be loaded part by part during the simulation. An editor is provided to create and maintain the descriptor data.

The unique feature of the proposed simulator is the integration of the dynamic jamming with MSE simulation. The jamming simulation is controlled by the jamming descriptor, which is a list of jamming specification tables in the order of their occurrence time. All tables have the same data structure. Each table is used in the time interval specified in the table. The tables have an entry for every radio unit. In each entry, the jammer can specify the frequency bands to be jammed and the jamming power. Note that different frequency bands can be jammed with different power. In this manner, the jammers can specify the time and duration of jamming, the radio units and frequency bands

to be jammed, and the jamming power to be used. To support the jamming simulation, an editor is provided to create and edit the jamming descriptor data. The data can be saved in a file, and it can be loaded into memory before the simulation if there is enough memory, or it can be loaded part by part during the simulation.

Channel simulation is the lowest level of the proposed MSE simulator. This layer contains appropriate routines for each type of radio link. A set of routines is used to compute the signal strength of each channel based on the jamming descriptor data. The results are used to determine whether the channels are jammed or not. This information will be used in the MSE radio protocol. Another set of routines is used to compute the error probabilities of the search, connection, end-of-routing, and status messages. The results are used by the flood search algorithm and affiliation process to determine whether the message is received correctly or not. Notice that no actual data are processed by the routines here. Either the signal strength and the message error probabilities are computed based on the mathematical models for the radio links. We will see later (in Section 3.2) that by providing another set of routines for this layer, we can come up with the channel emulator which is capable of processing the real data.

Implementation:

Many modules in the PRnet simulator can be used for the MSE simulator. First of all, the modules in the user interface layer are reusable. Our PRnet simulator supports only one type of node. In order to simulate a real network like MSE, the new simulator must support many types of nodes. Thus, enhancements are needed. The modules for dynamic jamming simulation are also reusable. Some modifications are needed to support the proposed object-oriented software architecture. To create the MSE simulator, we need the additional modules for the flood search algorithm, affiliation process, radio protocol, and channel simulation.

3.2 A MSE Emulator with Dynamic Jamming Emulation Capability

The MSE emulator is more complex than the simulator. The simulator is designed to run the complete MSE network on a computer. The emulator allows the use of multiple computers to emulate the MSE network. More precisely, the MSE network can be divided into pieces with each piece running on a computer. The hardware configuration of a emulated MSE network is illustrated in Figure 2. A computer can emulate a subnetwork containing one or more NC/LEN switches. We can also perform the channel emulation on the dedicated computers. The computers are connected via the standard asynchronous ports. The configuration of the subnetwork for a computer is specified in a configuration file. As discussed previously, every node is described by its corresponding object in the emulator. Thus, the subnetwork is emulated in terms of a set of objects. Since a NC/LEN switch on one computer may be connected to another NC/LEN switch on another computer via the asynchronous port, the configuration must also specify how the channels are emulated. If the asynchronous ports are used, the asynchronous communication driver will be used for data transfer between the computers. The asynchronous communication driver becomes an integrated component of the channel emulation. Note that the usage of the asynchronous communication driver is completely transparent to the higher layer network functions, which communicate with each other via the channel I/O buffers.

Software Architecture:

The architecture is illustrated in Figure 3. Notice that Figure 3 is similar to Figure 1 except the channel emulation layer contains one more component than the channel simulation layer shown in Figure 1. The additional component in the channel emulation layer is the asynchronous communication drivers, which are transparent to the higher layer functions.

All features described for the simulator are also available for the emulator. In additional, the emulator is designed to process the real data. Therefore, the routines in each layer of the emulator are more complex than their counterparts in the simulator.

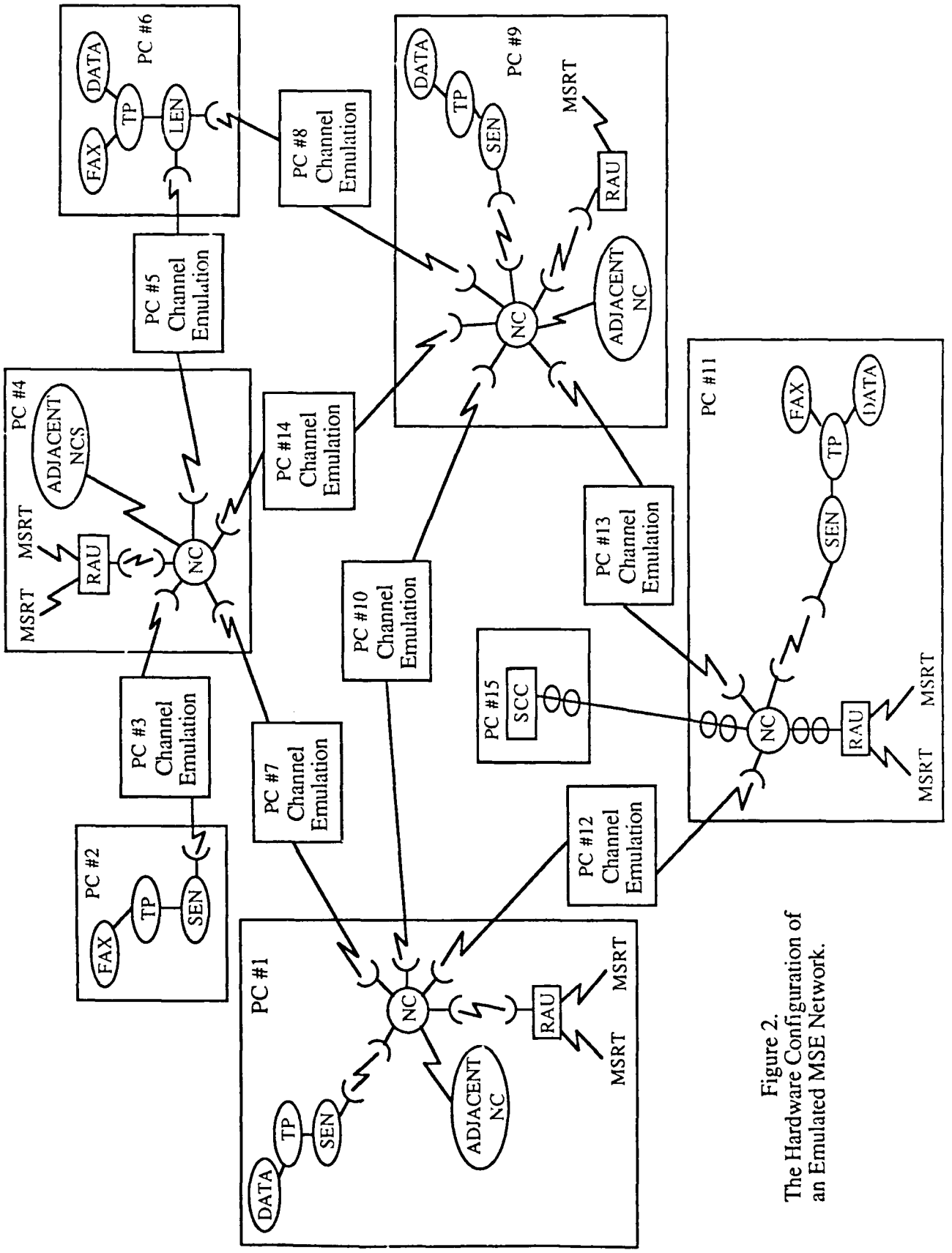

Figure 2.
The Hardware Configuration of an Emulated MSE Network.

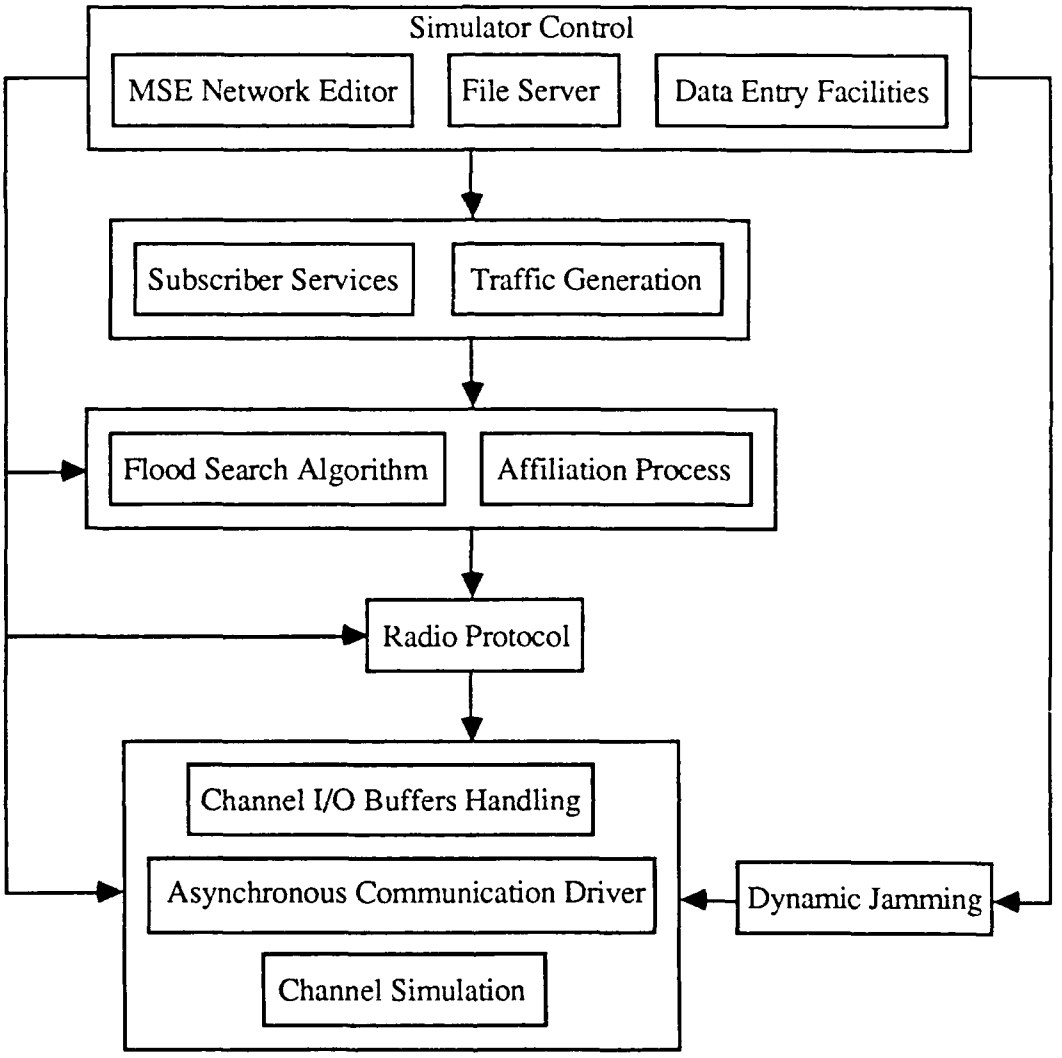

Figure 3. The Architecture of the MSE Simulator.

Function Descriptions:

To use the emulator, operators must prepare a configuration file, which defines the emulated subnetwork, for each computer. The configuration for a subnetwork must also specify its connectivity to the other subnetworks. To start an emulated MSE network, operators simply start every subnetwork emulation on its respective computer. The computers dedicated for channel emulation should be started as well. The computers are connected according to the prescribed topology. The emulated network will be established

by all component subnetworks automatically following the MSE network protocol. If there are any incorrect connections between computers, the corresponding links will be marked as used and be ignored. Operators can also start the desired input traffic emulation, affiliation emulation, and dynamic jamming emulation.

The call request from a subscriber terminal is forwarded to the NC/LEN switch using the real signal. The corresponding search message is then issued by the NC/LEN switch. The search message and the subsequent connection and end-of-routing messages contain the real routine data. The messages from the flood search algorithm are passed to the radio protocol layer. A proper channel is selected according to the radio protocol and the messages are queued in the corresponding channel input buffer. The channel simulation layer takes the data in the input buffer, processes them appropriately, and then passes the post-processing data to the output buffer. Note that the asynchronous communications driver will be used if it is needed during emulation. The radio protocol layer retrieves the data from the channel output buffers and passes them to the flood search algorithm. If a connection message is received, a circuit will be set up for the subscriber terminal.

Implementation:

Most of the software modules in the MSE simulator can be used for the MSE emulator. The functions, which generate message tokens in the simulator, must generate messages with real data in the emulator. The additional programming effort is not great because of the software architecture and data structures. In usual programming practice, there is a routine for generating each type of messages. The input of the message generation routines are the required information or data for the respective messages. The output of the message generation routines are message tokens for the simulator, and are real messages for the emulator.

The second main effort in upgrading the simulator to the emulator is to program the channel emulation layer. We have to add the asynchronous communication drivers and the symbol-by-symbol emulation capability to the emulator. The way to use the dynamic jamming data must also be changed appropriately. In this case, the jamming noise is added to the received symbol. The receivers try to detect the messages from the received symbols.

4. SUMMARY

In this report, we explain the software architecture and object-oriented programming methodology for a MSE simulator and emulator with the dynamic jamming simulation and emulation capabilities, respectively. We illustrate the way to extend our existing PRnet simulation for the MSE application. The incremental development concept is discussed in great detail. It can minimize the development cost as well as the risk. The product from each phase of development will be demonstrable and usable. The design of the simulator and emulator is aimed providing wide usage. The software will have a vary friendly user interface and a comprehensive context sensitive help system. The simulator is for performance evaluation of the MSE network and the emulator is for the network operation tests under the dynamic jamming attack.